In 2001, the 6th International Conference on Reliable Software Technologies will take place in Leuven, Belgium, from May 14th to May 18th. The conference offers a technical program and exhibition, plus a series of tutorials and a workshop. The conference provides an international forum for researchers, developers and users of reliable software technologies. Presentations and discussions cover applied and theoretical work currently conducted to support the development and maintenance of software systems.

The technical program includes keynote addresses, session papers by the international community with refereed contributions from many countries. The proceedings of the conference will be published in the Lecture Notes in Computer Science (LNCS) Series by Springer. An exhibition concurrent to the conference offers an opportunity to explore the latest developments by the commercial marketplace. The tutorials on Monday and Friday offer an excellent opportunity to obtain in-depth knowledge in important technologies in the field. Leuven is one of Europe's ancient university towns with many old beautiful buildings. The town hall, the famous Beguinage (Begijnhof), the churches, cloisters and colleges stand silent witness to the rich history of Leuven and make it the interesting place it is. Leuven is only 25 km (15 miles) away from Brussels, the capital of Belgium and the seat of the European Union. This year, K.U. Leuven university celebrates its 575th anniversary. The Conference will take place in the unique setting of the classicist buildings of Maria-Theresia College, situated in the historical centre of Leuven.

### Overview of the week

<table>
<thead>
<tr>
<th>Monday May 14th</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tutorials &amp; Workshop</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Non-standard techniques in Ada</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Practical experiences of safety-critical Ada technologies</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Early reliability measurement and improvement</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>An introduction to XML</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tuesday May 15th</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sessions &amp; Exhibition</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Building formal models for software requirements Axel van Lamsweerde, UCL, Belgium</td>
<td>Program Analysis</td>
<td>Distributed Systems</td>
<td>Using Ada in interactive digital television systems Pascal Heraud, CANAL+</td>
</tr>
<tr>
<td></td>
<td>Software Process</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vendor presentations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Wednesday May 16th</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sessions &amp; Exhibition</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Testing from formal specifications: a generic approach Marie-Claude Gaudel, Université de Paris-Sud</td>
<td>Real-Time Systems</td>
<td>Dependable Systems</td>
<td>APIs and Components</td>
</tr>
<tr>
<td></td>
<td>Language and Patterns</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vendor presentations</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vendor presentations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Thursday May 17th</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sessions &amp; Exhibition</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Standard Formats: UML &amp; XML</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Vendor presentations</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Friday May 18th</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Tutorials</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>From full concurrency to safe concurrency</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Building distributed systems with Ada</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Implementing design patterns in sequential Ada</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Architecture centered development and evolution of reliable real-time systems</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Exhibition

The exhibition opens in the mid morning break on Tuesday and runs until after the Thursday afternoon break. Each exhibitor will have at least one half hour presentation slot during the vendor track. The mid morning and mid afternoon breaks are one hour to give attendees ample opportunity to visit the exhibition.

At the time of writing five exhibitors - Aonix, ACT, TNI, Praxis Critical Systems and Irvine Compiler - have come forward and several others have expressed interest.

### Social Program

There are activities organised for four of the evenings starting with a welcome reception on Monday. Tuesday the City of Leuven have invited us all for a tour of the historic town hall followed by a reception. Wednesday there will be a guided walk through the historic Begijnhof followed by the conference banquet - be sure to book a ticket for this in your application. Thursday we have been invited to tour the Stella Artois brewery and enjoy a few of their world famous beers.
As the complexity of modern software systems grows, so does the need to deal reliably and efficiently with an increasing number of abnormal situations. The most general mechanism for this is exception handling, which is becoming a standard feature in modern languages.

A general exception handling mechanism should be well integrated with the other features of a language and conform to its programming paradigms. Increasing evidence from researchers and practitioners indicates that the exception handling in Ada 95 does not adequately reflect the whole range of programming paradigms supported by the language. In particular, the exception handling model remains based on Ada 83 while Ada 95 is object-oriented. Furthermore, exceptions and concurrency are, arguably, not well integrated. A task with an unhandled exception dies silently, and one has to resort to asynchronous transfer of control for passing exceptions asynchronously between tasks. It is not clear that this solution extends well into a distributed environment. Yet another problem is the existence of anonymous exceptions.

New fault tolerance schemes based on existing exception handling facilities have been developed in research environments. This is important as it allows higher level abstractions providing more advanced mechanisms to be introduced without impacting the language definition.

The aims of the workshop are:

• to share experience on how to build modern systems that have to deal with abnormal situations
• to discuss how solutions to those needs can be developed employing standard Ada features including the current exception handling paradigm
• to propose new exception handling mechanisms / paradigms that can be included in future revisions of the Ada language and also fit high integrity language profiles for safety critical systems.

Participation
Participation to the workshop is limited to 25-35 individuals and is by invitation upon acceptance of a submission. All types of submissions are welcome: brief position papers, experience reports, full research papers, etc. All papers will be made available to workshop participants before the workshop. The workshop will include talks based on the submitted papers and intensive shepherded discussion sessions. The submissions and a workshop summary will be published in Ada Letters.

Submissions
Submissions should be sent electronically (preferable in ps or pdf format) to Alexander Romanovsky:
alexander.romanovsky@ncl.ac.uk

Timetable
Electronic submission: Jan 31, 2001
Notification: Mar 15, 2001
Revised versions of papers: Apr 15, 2001

Workshop co-chairs:
Alexander Romanovsky (University of Newcastle)
Alfred Strohmeier (EPFL)
Andy Wellings (University of York)

Workshop Program Committee:
Bill Bail (MITRE)
Jörg Kienzle (EPFL)
Pat Rogers (Software Arts and Sciences)
Bo Sanden (Colorado Technical University)
Anand Tripathi (University of Minnesota)
Tullio Vardanega (ESA)
Thomas Wolf (Paranor)

Website
There is a link to the workshop website URL below from the conference website. See Program → workshops → workshop web page.

http://www.cs.ncl.ac.uk/people/alexander.romanovsky/home.formal/eh-ada.html

### Tutorials

<table>
<thead>
<tr>
<th>Monday May 14th</th>
<th>Friday May 18th</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Non-standard techniques in Ada</strong></td>
<td><strong>Implementing design patterns in sequential Ada</strong></td>
</tr>
<tr>
<td>break</td>
<td>break</td>
</tr>
<tr>
<td><strong>Practical experiences of safety-critical Ada technologies</strong></td>
<td><strong>Building distributed systems with Ada</strong></td>
</tr>
<tr>
<td>T1 Art Duncan</td>
<td>T6 Samuel Tardieu, Laurent Pautet, Thomas Quinot</td>
</tr>
<tr>
<td>lunch</td>
<td>T7 Matthew Heaney</td>
</tr>
<tr>
<td><strong>Early reliability measurement and improvement</strong></td>
<td><strong>From full to safe concurrency</strong></td>
</tr>
<tr>
<td>09:00</td>
<td>lunch</td>
</tr>
<tr>
<td>10:30</td>
<td>lunch</td>
</tr>
<tr>
<td>11:00</td>
<td>lunch</td>
</tr>
<tr>
<td>12:30</td>
<td>lunch</td>
</tr>
<tr>
<td>14:00</td>
<td>lunch</td>
</tr>
<tr>
<td>15:30</td>
<td>lunch</td>
</tr>
<tr>
<td>16:00</td>
<td>lunch</td>
</tr>
<tr>
<td>17:30</td>
<td>lunch</td>
</tr>
<tr>
<td><strong>From full concurrency to safe concurrency</strong></td>
<td><strong>Architecture centered development and evolution of reliable real-time systems</strong></td>
</tr>
<tr>
<td>break</td>
<td>break</td>
</tr>
<tr>
<td><strong>Practical experiences of safety-critical Ada technologies</strong></td>
<td><strong>Building distributed systems with Ada</strong></td>
</tr>
<tr>
<td>T2 Peter Amey, Rod Chapman</td>
<td>T8 Bruce Lewis, Ed Colbert</td>
</tr>
<tr>
<td>T3 Jeff Tian</td>
<td></td>
</tr>
<tr>
<td>T4 Gregory Neven</td>
<td></td>
</tr>
<tr>
<td><strong>An introduction to XML</strong></td>
<td><strong>From full to safe concurrency</strong></td>
</tr>
<tr>
<td>break</td>
<td>break</td>
</tr>
<tr>
<td><strong>Implementing design patterns in sequential Ada</strong></td>
<td><strong>Building distributed systems with Ada</strong></td>
</tr>
<tr>
<td>break</td>
<td>break</td>
</tr>
</tbody>
</table>

### Monday May 14th

- **T1** Art Duncan
- **T2** Peter Amey, Rod Chapman
- **T3** Jeff Tian
- **T4** Gregory Neven

**Lunch**

**Non-standard techniques in Ada**

- break

**Practical experiences of safety-critical Ada technologies**

- break

**Early reliability measurement and improvement**

- break

**From full concurrency to safe concurrency**

- break

Refreshments will be served in the breaks. Lunch is included for those following all-day programs or two half day tutorials on the same day. Additional lunch tickets are on sale throughout the conference.
Non-standard techniques in Ada
Art Duncan, RPI, USA.
Monday May 14th, all day T1

The tutorial is directed toward students and educators as well as professional Ada programmers. The techniques presented have the dual advantage of being both useful and fun.

We will go beyond the normal introductory course in Ada to explore a number of interesting non-standard ways of using Ada. Many of the techniques have been inspired by the author’s experiences with the programming paradigms used by programmers of Lisp and various functional programming languages.

The tutorial shows how we can reduce complexity and improve readability of Ada programs by using a number of declarative techniques to replace potentially complex and involved procedural code. These techniques include:

- declarative techniques for encapsulating control structures,
- a flexible, data-driven technique that, for want of a better term, I have called “template-driven programming,”
- use of inheritance and iterators to define generalized applications, and
- a technique based on lazy evaluation for the creation and manipulation of infinite data structures.

The tutorial will last a full day and will consist of five sections, followed by the author’s conclusions and suggestions for further investigation.

Presenter
Dr. Arthur G. Duncan is an independent consultant, as well as an adjunct professor of Computer Science at the Rensselaer Polytechnic Institute in Troy, New York, U.S.A.

Dr. Duncan has been involved with the Ada language for 20 years. Before becoming an independent consultant, he was a key member of the Ada group at the G.E. Research Laboratory, and later at the Rensselaer Polytechnic Institute.

Dr. Duncan obtained a Ph.D. from the University of Michigan and a Master of Science from the Rensselaer Polytechnic Institute.

Practical experiences of safety-critical Ada technologies
Peter Amey & Rod Chapman, Praxis Critical Systems, UK.
Monday May 14th, all day T2

The tutorial identifies the special properties of systems intended for use in ultra-reliable domains and the qualitative shift in development methods that is required to achieve those properties. The advantages (and weaknesses) of Ada are introduced in the context of the ISO HRG report on High-Integrity Ada and the SPARK sub-language. The demands of common, important development standards are described together with appropriate and cost-effective techniques for meeting them. Finally project experience illustrating successes in meeting the main standards is discussed.

Presenters
The tutorial will be presented by Mr Peter Amey and Dr Rod Chapman of Praxis Critical Systems. Both have extensive teaching and presentation experience mostly obtained in the subject field. Both teach regularly on the ‘Software Engineering with SPARK’ course run by Praxis Critical Systems. Dr Chapman has been a regular presenter at Ada Europe, Ada Sweden, SIGAda and the World Congress on Formal Methods.

Early reliability measurement and improvement
Jeff Tian, SMU, USA
Monday May 14th, morning T3

This tutorial introduces basic concepts in software reliability engineering (SRE) and surveys recent developments in SRE.

Existing reliability models, including both the time domain software reliability growth models (SRGMs) and input domain reliability models based on repeated random sampling are surveyed. Both these types of models were integrated in the recently developed tree-based reliability models (TBRMs) by the author to analyze product reliability and identify high risk areas for focused reliability improvement. Various practical issues in applying existing SRE techniques and the new TBRMs in large software systems are also discussed, including: environmental constraints, measurement types and availability, SRE implementation strategies, and support tools for analysis, modeling, and SRE implementations.

This approach has been used in the testing phase of several large software products developed in the IBM Software Solutions Toronto Laboratory and was demonstrated to be effective and efficient. In addition, other recent work in early measurement of software reliability, integration of SRE and traditional software measurement and analysis activities, and analysis of techniques for reliability assurance and improvement are also covered in this tutorial.

Presenter
Jeff presented a similar tutorial at Ada-Europe 2000 entitled “Tree-Based Reliability Models (TBRMs) for Early Reliability Measurement and Improvement”. The tutorial was rated the best by the attendees. Jeff was ranked as one of the “Top Scholars in Systems and Software Engineering” (only top 15 or so ranked) by the Journal of Systems and Software in 1997, 1998, and 1999.

Jeff obtained a Ph.D. from the University of Maryland in 1992 then worked for IBM until 1995 when he became Assistant Professor of Computer Science and Engineering at Southern Methodist University (SMU) since Fall, 1995. Since then Jeff has developed and taught two closely related courses entitled CSE 8317/8391 Software Reliability and Safety and CSE 6340 Advanced Topics in Software Engineering in 1996-2000 at SMU.

An introduction to XML
Gregory Neven, K.U.Leuven, Belgium
Monday May 14th, afternoon T4

The Extensible Markup Language (XML) has emerged in just a few years as nothing less than a phenomenon in computing. XML is a “meta language”: it is a standard for defining descriptions of structure and content in documents. If HTML provides a universal method to display data, XML takes the next step by separating the content of a document from its representation and by providing not only the data itself but also its interpretation. XML transforms data into information. This universal, flexible and extensible approach opens up an almost unlimited range of uses for XML, from word processing through e-business to data archiving.

Outline
Discussed subjects include:
- The XML syntax
- How to define constraints on XML documents (DTD’s, Schema)
- How to parse, traverse and transform XML documents (SAX, DOM, XSLT)
- Some XML standards like ebXML, Biztalk, SOAP, ...
- An example that illustrates XML’s role in B2B e-commerce

Presenters:
Gregory Neven, Maarten Coene, Roel Adriaensens are all teaching assistants at K.U. Leuven with extensive XML experience.

From full concurrency to safe concurrency
John Harbaugh, Boeing, USA
Friday May 18th, all day T5

The outlook for embedded systems has never been more exciting. Fast 32-bit processors and cheap memory are fueling an explosion of sophisticated applications. From “smart cards” for financial, medical, and governmental records, to drive-by-wire automobiles and integrated e-phones, many “mission-critical” aspects of our lives are coming to depend on the software running on all those processors. Recent research has led to a compact and reliable tasking model, the so-called Ravenscar profile, designed specifically for high-integrity, efficient, real-time systems. In this tutorial, students will learn how to write concurrent (multithreaded) applications that are consistent with the Ravenscar profile and use the Ada95 programming language. Students will also learn how to assure that real-time programs will meet their deadlines using Rate Monotonic Analysis.

Tutorials
Outline
This tutorial introduces Ada95's support for concurrent program design. Students will explore Ada tasking in the context of high integrity applications suitable for safety/mis- sion critical computing.
Language features are introduced in the context of the Ravenscar profile for high integrity Ada tasking. Students will learn to analyze task schedulability and estimate CPU utilization using Rate Monotonic Analysis.

Presenter
John Harbaugh graduated with a BSEE from the University of Washington in 1980 and has worked for The Boeing Company ever since. John has written and tested embedded software for missiles, helicopters, and commercial aircraft, much of it in Ada and assembler. From 1995 to 1999 he was on staff in the Employee Training & Development organization where he developed and taught courses in software engineering, using Ada as the primary teaching language. He currently writes software for a large, distributed, Object Ori- ented upgrade program to the NATO Airborne Warning and Control System (AWACS) using Ada95.

Building distributed systems with Ada
Samuel Tardieu, Laurent Pautet & Thomas Quinot, ENST, France
Friday May 18th, all day T6
This tutorial will cover the various ways of conceiving and building a distributed application in Ada95. From BSD sockets to the Dis- tributed Systems Annex, the audience will learn how to efficiently and easily build a powerful distributed application using portable methods.

Description of the topic
Every month subscribers to new software mailing-lists see a new language specially designed for the Internet, whose revolution- nary communication capabilities will solve any problem encountered by designers of distributed applications. Unfortunately, these language are often short-lived and are solely used by their authors. However, Ada is a long-lived language whose communication skills are hardly known. Since Ada 95, bindings to standard communication services make it easy to use any network layer. Moreover, the standardization of CORBA for Ada and of the Distributed Sys- tems Annex make it one of the most easy-to-use languages for building large distributed applications.
This tutorial will show how Ada can be used efficiently to build reliable distributed applications. Although proprietary software also exists, the tutorial will outline standardized solutions that are available as free software packages, so that the audience can experi- ment with them right after the course.

Implementing design patterns in sequential Ada
Matthew Heaney, USA
Friday May 18th, morning T7
In this tutorial I discuss design pattern technol- ogy, with a focus on the implementation issues likely to confront the Ada95 program- mer.

Outline
Interpreter Pattern
• Features idiom for class-wide programming and memory management.
• Flyweight Pattern; Template Methods.
Smart Pointers
• Memory management technique (using control- led types) to eliminate storage leaks.
Factory Method
• Solves the problem of how to efficiently copy a stack having a class-wide type.
Observer Pattern
• Observing one subject; observing multiple sub- jects (the "multiple views" idiom); observing an observer; observing specific attributes so you know what has been changed.
Rosen Trick
• Legal and efficient way to modify an object passed as an in-mode subroutine parameter; show how to implement function Random.
Generic Dispatching
• Dispatch a generic formal subprogram; extend idea to allow dispatching through an access-to-
subprogram object.

Presenter
I have been programming in Ada for twelve years (mostly building real-time systems), and spent a couple of years converting all the C++ examples in the Gamma book to Ada95.

Architecture centred development and evolution of reliable real-time systems
Bruce Lewis & Ed Colbert, USA
Friday May 18th, afternoon T8
This tutorial will describe an architecture cen- tric approach to developing and evolving reli- able systems using the MetaH Architecture Description Language and UML. MetaH is an ADL specifically developed for time critical, reliable applications. It supports building multiprocessor, fault tolerant, multi-level safety critical systems. It was developed for the avionics domain but useful in many appli- cations such as robotics, simulation, engine controls, automotive etc. UML is a widely used object orient general purpose software specification language. The tutorial will describe its specialization mapping and use with the real-time ADL MetaH. It will also briefly describe standardization activities of the Avionics Architecture Description Lan- guage under SAE based as well as new capabilities and research directions.

Description of the topic:
We believe that the most powerful approach to building real-time reliable, evolvable architec- ture centered systems is through an architec- ture description language that supports spec- ification of the architecture, modeling of architecture level system properties and auto- mated integration of hardware and software components from multiple sources preserving modeled properties. Glue code is generated in conformance to specification. Generated kernel supports space and time partitioning for multi-level safety and reduced testing and validation costs. Generation is in Ada95 and supports the transparent integration of Ada, C, C++ software components and use of the Ada95 runtime or POSIX. The approach reduces costs in multiple significant areas in software development and evolution - system level design error reduction, rapid system integration and re-integration, code genera- tion for complex component infrastructure, rapid retargeting of time critical software to new execution environments, and reduced revalidation costs. Extensions for UML for real-time architecture description are needed by and are of high interest to industry and at least an hour will be devoted to this topic in the tutorial.

Presenter
Bruce Lewis
Technical POC for DARPA MetaH develop- ment over last 7 years Chair of Architecture Description Language task group for SAE, Avionics Software Engineering Directorate Lead for advanced software technology development and acquisition.
Ed Colbert
Taught and consulted on software engineer- ing, including object-oriented methods and languages, Ada and UML since 1982. Creator of the Colbert ObjectOriented Soft- ware Development method (OOSD), which NASA Langley Research Center used for a Software Engineering Process manual, choosing OOSD partly for its strength in real- time software development.
Developer of UML Real Time ADL modeling capability for MetaH
Co-taught tutorial on MetaH/UML at the TOOLS USA 2000 Conference.
# Tuesday May 15th

**Invited speech:**
**Building formal models for software requirements**
Axel van Lamswaarde, UCL, Belgium

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:00</td>
<td>break - exhibition opens</td>
</tr>
<tr>
<td>10:00</td>
<td>Program Analysis</td>
</tr>
<tr>
<td>11:00</td>
<td>Software Process</td>
</tr>
<tr>
<td>11:30</td>
<td>Parameter-induced aliasing in Ada</td>
</tr>
<tr>
<td>11:45</td>
<td>Slicing tagged objects in Ada</td>
</tr>
<tr>
<td>12:00</td>
<td>OASIS - an ASIS secondary library for analyzing object-oriented Ada code</td>
</tr>
<tr>
<td>12:30</td>
<td>lunch &amp; exhibition</td>
</tr>
<tr>
<td>13:30</td>
<td>Real - Time Systems</td>
</tr>
<tr>
<td>13:45</td>
<td>New developments in Ada runtime profile definition and language refinements</td>
</tr>
<tr>
<td>14:00</td>
<td>Experience templates in Ada</td>
</tr>
<tr>
<td>14:30</td>
<td>Vendor presentation</td>
</tr>
<tr>
<td>15:00</td>
<td>Defining new non-preemptive dispatching and locking policies for Ada</td>
</tr>
<tr>
<td>15:30</td>
<td>lunch &amp; exhibition</td>
</tr>
<tr>
<td>16:30</td>
<td>System Evolution</td>
</tr>
<tr>
<td>16:45</td>
<td>Vendor presentation</td>
</tr>
<tr>
<td>17:00</td>
<td>Ship System 2000, a stable architecture under continuous evolution</td>
</tr>
<tr>
<td>17:15</td>
<td>Vendor presentation</td>
</tr>
<tr>
<td>18:00</td>
<td>Ada-Europe General Assembly</td>
</tr>
</tbody>
</table>

**Vendor presentation**

- Building modern distributed systems
- A toolkit for reliable communication in distributed computer-controlled systems
- Building robust applications by reusing non-proprietary software
- Experience report: using the SPARK toolset for showing the absence of run-time errors in safety-critical software
- Scenario-based system assessment
- Test suite reduction and fault detection effectiveness: an empirical evaluation

**Invited speech:**
**Testing from formal specifications: a generic approach**
Marie-Claude Gauzel, Université de Paris-Sud

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>09:00</td>
<td>break - exhibition opens</td>
</tr>
<tr>
<td>10:00</td>
<td>Real - Time Kernels</td>
</tr>
<tr>
<td>11:00</td>
<td>MaRTE OS: an Ada kernel for real-time embedded applications</td>
</tr>
<tr>
<td>11:30</td>
<td>Implementing Ada.Real.Time.Clock and absolute delays in real time kernels</td>
</tr>
<tr>
<td>12:00</td>
<td>Component libraries and language features</td>
</tr>
<tr>
<td>12:30</td>
<td>Defining new non-preemptive dispatching and locking policies for Ada</td>
</tr>
<tr>
<td>13:30</td>
<td>Architecture Crosses</td>
</tr>
<tr>
<td>13:45</td>
<td>Vendor presentation</td>
</tr>
<tr>
<td>14:00</td>
<td>Mapping UML to Ada</td>
</tr>
<tr>
<td>14:30</td>
<td>Vendor presentation</td>
</tr>
<tr>
<td>15:00</td>
<td>An application case for the Ravenscar Profile</td>
</tr>
</tbody>
</table>

**Invited speech:**
**Logic versus Magic in critical systems**
Peter Amey, Praxis Critical Systems, UK

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>18:00</td>
<td>Guided tour of historic town hall</td>
</tr>
<tr>
<td>18:15</td>
<td>break - exhibition final session</td>
</tr>
<tr>
<td>19:00</td>
<td>Banquet at Faculty Club in Groot Begijnhof</td>
</tr>
<tr>
<td>20:00</td>
<td>Visit to Stella Artois brewery</td>
</tr>
</tbody>
</table>

### Program Analysis

- Parameter-induced aliasing in Ada
  - Wolfgang Gellerich & Erhard Pödereder
  - Why we have problems producing quality software: When the product and process become confused
  - David A. Cook & Les Dupax

- Slicing tagged objects in Ada
  - Z. Chen, B. Xu & H. Yang
  - The importance of agreed software quality metrics
  - Jens Pas

- OASIS - an ASIS secondary library for analyzing object-oriented Ada code
  - Alexei Kuchumov, Sergey Rybin & Alfred Strohmeier

### Software Process

- New developments in Ada runtime profile definition and language refinements
  - Joyce L. Tokar

- A design pattern for state machines and concurrent activities
  - Bo Sanden

- Implementing a flexible scheduler in Ada
  - Guillem Bernal & Alan Burns

### Dependable Systems

- Building modern distributed systems
  - Laurence Pauler & Samuel Tardieu
- A toolkit for reliable communication in distributed computer-controlled systems
  - Luis Miguel Pinho & Francisco Vasques
- Building robust applications by reusing non-proprietary software
  - F. Guerra, J. Miranda & J. Calero

- Experience report: using the SPARK toolset for showing the absence of run-time errors in safety-critical software
  - Darren Foulger & Steve King

- Scenario-based system assessment
  - Sille Kudahl et al

- Test suite reduction and fault detection effectiveness: an empirical evaluation
  - T.Y. Chen & M.F. Lau

### APIs and Components

- JEWL: a GUI library for educational use
  - John English
- Object-oriented stable storage based on mirroring
  - Xavier Caron, Jörg Kienzle & Alfred Strohmeier
- Transaction support for Ada
  - Jörg Kienzle, Ricardo Jiménez Pens, Alexander Romanovsky & M. Patih Martinez

**Vendor presentation**

- A design pattern for state machines and concurrent activities
- A toolkit for reliable communication in distributed computer-controlled systems
- Building robust applications by reusing non-proprietary software
- Building modern distributed systems
- Experience report: using the SPARK toolset for showing the absence of run-time errors in safety-critical software
- Scenario-based system assessment
- Test suite reduction and fault detection effectiveness: an empirical evaluation
- Architecture Crosses
- An application case for the Ravenscar Profile: Porting OBOSS to GNAT/ORK

**Invited speech:**
**Using Ada in interactive digital television systems**
Pascal Heraud

<table>
<thead>
<tr>
<th>Time</th>
<th>Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>17:00</td>
<td>Vendor presentation</td>
</tr>
</tbody>
</table>

**Vendor presentation**

- Can Java meet its real-time deadlines?
  - Brian Dobbing, Aonix Europe Ltd.
  - co-author Ben Brosgol, ACT.
Invited Speakers

Building formal models for software requirements
Axel van Lamsweerde, Université Catholique de Louvain, Belgium

Tuesday May 15th, 09:00
Requirements engineering (RE) is concerned with the elicitation of the goals to be achieved by the system envisioned, the operationalization of such goals into specifications of services and constraints, and the assignment of responsibilities for the resulting requirements to agents such as humans, devices, and software. Getting high-quality requirements is difficult and critical. Recent surveys have confirmed the growing recognition of RE as an area of primary concern in software engineering research and practice.

The talk will first briefly introduce RE by discussing its main motivations, objectives, activities, and challenges. The role of rich models as a common interface to all RE processes will be emphasized.

We will then review various techniques available to date for system modeling, from semi-formal to formal, with the aim of showing their relative strengths and weaknesses when applied during the RE stage of the software lifecycle, notably, their limited scope, their lack of abstraction, their poor separation of concerns, and their lack of methodological guidance.

The talk will then discuss a number of recent efforts to overcome such problems through RE-specific techniques for goal-oriented elaboration of requirements, multiparadigm specification, the handling of non-functional requirements, the management of conflicting requirements, and the handling of abnormal agent behaviors.

Short biography
Axel van Lamsweerde is Full Professor of Computing Science at the University of Louvain, Belgium. He is co-founder of the CEDITI technology transfer institute partially funded by the European Union. He has also been a research associate at Philips Research Laboratories, the University of Oregon, and the Computer Science Laboratory of SRI International, Menlo Park, CA. His professional interests are in lightweight formal methods for reasoning about software engineering products and processes.

Axel van Lamsweerde is an ACM fellow. Since 1995, he is Editor-in-Chief of the ACM Transactions on Software Engineering and Methodology (TOSEM). He has been program chair of major software engineering conferences and workshops, including ESEC’91, ICSE’94, and IWSSD’93. He is member of the Editorial Boards of the Automated Software Engineering Journal and the Requirements Engineering Journal.

Using Ada in interactive digital television systems
Pascal Heraud, CANAL+ Technologies, France

Tuesday May 15th, 16:30
The digital television (DTV) market has been growing exponentially since 1996. Based on widely accepted MPEG and DVB standards, digital television offers a higher image quality as well as an unlimited number of interactive services.

Canal+ Technologies provides a complete end-to-end solution for digital television operators, from the central broadcast centers to the set-top boxes at home. The DTV broadcast center systems have availability, reliability and load constraints which require a robust implementation.

For this reason, the server-side components of Canal+ Technologies software have been developed in Ada.

This presentation explains the architecture of a digital television system and how Ada is used inside this system. It also describes how such a system is currently re-engineered from a proprietary Ada 83 / OpenVMS implementation using the DEC Ada compiler to an Ada 95 multi-platform implementation using the GNAT compiler.

Short biography
Pascal Héraud works in the team doing the porting and re-engineering of the Canal+ applications from Ada 83 on OpenVMS to Ada 95 on multiple platforms. Before joining Canal+, he spent many years at Aonix as a software engineer working on both AdaWorld and ObjectAda products, both in Paris and San Diego.

Testing from formal specifications: a generic approach
Marie-Claude Gaudel, Université de Paris-Sud, France

Wednesday May 16th, 09:00
Deriving test cases from specifications is now recognised as a major application of formal approaches to software development.

Several solutions have been proposed for various formalisms: behavioural descriptions such as transition systems, model-based specifications, algebraic specifications, etc.

This talk will present our general approach of test data selection from formal specifications. A notion of “exhaustive test set” is derived from the semantics of the formal notation and from the definition of a correct implementation. Then a finite test set is selected via some “testing hypothesis”.

This approach will be illustrated by its application to the case of algebraic specifications, object oriented Petri nets (CO-OPN2), LUSTRE, and full Lotos.

Several case studies and industrial experiments will be reported.

Short biography
Marie-Claude Gaudel was appointed full professor at the University of Paris-Sud in 1984 and is now professeur classe exceptionelle. Before joining UPS, she was a researcher at INRIA, then managed the Software Engineering group at the industrial research center of Alcatel-Alsthom (Marcoussis, France).

Her research interests include formal methods, program robustness and testing methods. For many years, she has pushed for effective use of formal methods in all phases of critical system development, with an emphasis on testing activity.

She is president of the Scientific Board of INRIA and is chair of the Board of RENATER (the French computer network for education and research).

She is Doctor Honoris Causa of EPFL, and she got the CNRS Silver Medal in 1996 for her work on software testing.

Logic versus Magic in Critical Systems
Peter Amey, Praxis Critical Systems, UK

Thursday May 17th, 09:00
A prevailing trend in software engineering is the use of tools which apparently simplify the problem to be solved. Often, however, this results in complexity being concealed or "magicked away". For the most critical of systems, where a credible case for safety and integrity must be made prior to there being any service experience, we cannot tolerate concealed complexity and must be able to reason logically about the behaviour of the system.

The presentation draws on real life project experience to identify some historical and current magics and their effect on high integrity software development; this is contrasted with the profound quality benefits that can be made from taking a more logical and disciplined approach.

Short biography
Peter Amey is an aeronautical engineer by original professional training. He was a serving engineering officer in the Royal Air Force where he spent several years at the Boscombe Down test establishment working on the certification of aircraft armament systems. Peter joined Program Validation Ltd to develop SPARK and the SPARK Examiner and continues that work with Praxis Critical Systems. As well as developing SPARK he has used it on the Tornado, Eurofighter and Lockheed C130J programmes.

Can Java meet its real-time deadlines?
Brian Dobbing, Aonix Europe Ltd. co-author Ben Brosogl, ACT.

Thursday May 17th, 16:30
Ada has been-there, done-that as regards meeting real-time programming require-
mements. The Ada95 revision addressed almost all the concerns that had plagued Ada83's usability. But Java is now the flavor of the month for just about everything it seems. Current Java semantics for all things concurrent are much inferior to even the generally rejected Ada83 tasking model, and so two ongoing competing initiatives to fix Java concurrency are in progress. Both attempt to make Java suitable for real-time by addressing predictability, performance, footprint and missing features. But how successful are these attempts, and will they achieve the goal of producing high-reliability, high-performance and predictable Java-based software?

Short biography of presenter
Brian Dobbing is Chief Technical Consultant at Aonix Europe and has been involved in the production of Ada development tools and runtime systems for almost 20 years. He was a member of ISO WG9 during the Ada95 revision process and spearheaded the definition of the Ravenscar Profile. Brian is also technical editor of the J Consortium working group that is defining ISO standard extensions to the Java platform for high integrity systems.

Short biography of co-author
Ben Brosgol, a senior member of the technical staff of Ada Core Technologies in the U.S., has had a long and direct participation in the Ada effort as a language designer, implementor, educator and user. A well-known figure in the international Ada community and currently the Chair of ACM SIGAda, he has delivered many papers and tutorials at Ada conferences, including several in-depth comparisons of Ada and Java. Since early 1999 he has been involved with the real-time Java efforts. Ben is a primary member of the Real-Time for Java Expert Group, who have been operating under Sun's Java Community Process to specify a set of real-time extensions to the Java platform. He is also a reviewer of the J-Consortium's proposed core real-time Java extensions.
Family name______________________________________________ First name______________________________________________

Affiliation / Organisation _______________________________________________________________________________________

Address ___________________________________________________________________________________________________
________________________________________________________________________________________________________

City__________________________ Post / Zip code__________________________ Country _________________________________________

Telephone________________________ Fax ___________________________ E-mail__________________________

Accompanying Person(s) ______________________________________________________

All rooms with bath or shower. RATES PER ROOM PER NIGHT, BUFFET BREAKFAST & TAXES INCLUDED

<table>
<thead>
<tr>
<th>Hotel</th>
<th>Code</th>
<th>Single Room</th>
<th>Double Room</th>
</tr>
</thead>
<tbody>
<tr>
<td>Holiday Inn Garden Court*</td>
<td>LVHHOLI</td>
<td>4.250 BEF</td>
<td>4.250 BEF</td>
</tr>
<tr>
<td>Begijnhof Congreshotel*</td>
<td>LVHBEGI</td>
<td>5.400 BEF</td>
<td>5.900 BEF</td>
</tr>
<tr>
<td>Hotel Binnenhof *</td>
<td>LVHBINH</td>
<td>3.800 BEF</td>
<td>4.250 BEF</td>
</tr>
<tr>
<td>Hotel New Damshire</td>
<td>LVHNEWD</td>
<td>3.800 BEF</td>
<td>4.300 BEF</td>
</tr>
<tr>
<td>Hotel Ibis *</td>
<td>LVHIBIS</td>
<td>3.300 BEF</td>
<td>3.600 BEF</td>
</tr>
</tbody>
</table>

* = Parking space (no advance reservations) 1 EUR = 40.3399 BEF

First choice :______________________________________________ Second choice :______________________________________________

Arrival date : _____________________________ Departure date : ____________________ Number of nights : __________________

☑ SINGLE ROOM ☑ DOUBLE ROOM

Rooms will be allocated on a first-come-first-served basis. We strongly advise all participants to book early, as May is a busy period for hotels in Leuven.

PAYMENT

The total amount must be paid directly to the hotel when checking out. Reservations must be guaranteed by means of a Credit Card (Eurocard / Mastercard - Visa Card - Diners Club) or by a cheque (Bank- or Eurocheque) for an amount equivalent to 1 night.

I will pay by: Credit Card ☑ Credit Card Number __________________________ Bank / Eurocheque ☑

In case of cancellation less than 7 days before arrival or "no-show" (reservation not cancelled before arrival) 1 night will be charged to your credit card or your cheque will not be given back.

RESERVATIONS WILL ONLY BE ACCEPTED WITH CREDIT CARD GUARANTEE OR WITH ENCLOSED CHEQUE.

Send form to
OMNIA CONGRESS DEPT.
Parijsstraat 50
B-3000 LEUVEN (BELGIUM)
FAX +32 (0)16 243802
E-mail : gaelle.bourgault@omnia.be
REGISTRATION FORM

PARTICIPANT

Ms  Mr  __________

Family name__________________________________________ First name__________________________________________

Affiliation / Organisation _______________________________________________________________________________________

Address ___________________________________________________________________________________________________

________________________________________________________________________________________________________

City ____________________________________ Post / Zip code _____________________ Country___________________________

Telephone_______________________________ Fax ______________________________ E-mail____________________________

Special requirements (e.g. diet) ____________________________________________________________

Reduced registration fee

☐ member Ada-Europe; national organization _________  ☐ academica

☐ member ACM; membership number ______________

Registration Time

☐ Early registration (by April 12th)  ☐ Late or on site (after April 12th)

REGISTRATION FEES

Conference registration fee (see table on next page)

Three day conference ................................................................................................................................ EUR_________

One day (Tue ☐ Wed ☐ Thu ☐) ................................................................................................................ EUR_________

Tutorial registration (see table on next page)

Please indicate tutorials for which you want to register:

 Monday, May 14th    ☐ T 1    ☐ T 2    ☐ T 3    ☐ T 4

 Friday, May 18th    ☐ T 5    ☐ T 6    ☐ T 7    ☐ T 8

Tutorial registration fee ................................................................................................................................ EUR_______

Conference Dinner: _______tickets @ 55 EUR ........................................................................................................ EUR_______

Extra proceedings: _______proceedings @ 30 EUR ............................................................................................... EUR_______

TOTAL PAYMENT DUE ........................................................................................................................................ EUR_______

PAYMENT METHOD

account is at the KBC Bank, Belgium whose bank identifier (swift) code is KREDBEBB (Please mention “Ada-Europe 2001” and
your name and attach proof of payment, e.g. a copy of the bank draft, to this form).


☐ By credit card ☐ MasterCard ☐ Visa

Card # ___________________________ Expiration Date ___________________________

Name as shown on credit card ___________________________ Signature ___________________________

Mail or fax this form to: AE2001 Registration, Department of Computer Science, KULeuven, Celestijnenlaan 200A, B-3001 Leuven, Belgium.

Fax +32 (0)16 327996
Conference registration fee:

3 days of conference (May 15th – May 17th) including one copy of the proceedings, coffee breaks, lunches and welcome reception on Monday evening May 14th, visit and reception in town hall on Tuesday 15th, visit to Stella Artois brewery on Thursday 17th.

<table>
<thead>
<tr>
<th></th>
<th>member Ada-Europe or ACM</th>
<th>non member</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>non academia</td>
<td>academia</td>
</tr>
<tr>
<td>Early registration (by April 12th)</td>
<td>450 EUR</td>
<td>390 EUR</td>
</tr>
<tr>
<td>Late/on site registration (after April 12th)</td>
<td>510 EUR</td>
<td>510 EUR</td>
</tr>
<tr>
<td>One day registration</td>
<td>255 EUR</td>
<td>255 EUR</td>
</tr>
</tbody>
</table>

Tutorial registration fee:

Prices are per tutorial, including tutorial notes and coffee breaks. Lunches are only included when registered for full day tutorial or 2 half day tutorials on the same day.

<table>
<thead>
<tr>
<th></th>
<th>half day</th>
<th>full day or two halves on same day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Early registration (by April 12th)</td>
<td>120 EUR</td>
<td>230 EUR</td>
</tr>
<tr>
<td>Late/on site registration (after April 12th)</td>
<td>150 EUR</td>
<td>290 EUR</td>
</tr>
</tbody>
</table>

Overview of Tutorials:

<table>
<thead>
<tr>
<th>Date</th>
<th>Time</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monday May 14th</td>
<td>T1</td>
<td>full day Non-standard techniques in Ada - Art Duncan</td>
</tr>
<tr>
<td></td>
<td>T2</td>
<td>full day Practical experiences of safety-critical Ada technologies - Peter Amey &amp; Rod Chapman</td>
</tr>
<tr>
<td></td>
<td>T3</td>
<td>morning Early reliability measurement and improvement - Jeff Tian</td>
</tr>
<tr>
<td></td>
<td>T4</td>
<td>afternoon An introduction to XML - Gregory Neven</td>
</tr>
<tr>
<td>Friday May 18th</td>
<td>T5</td>
<td>full day From full concurrency to safe concurrency - John Harbaugh</td>
</tr>
<tr>
<td></td>
<td>T6</td>
<td>full day Building distributed systems with Ada - Samuel Tardieu, Laurent Pautet, Thomas Quinot</td>
</tr>
<tr>
<td></td>
<td>T7</td>
<td>morning Implementing design patterns in sequential Ada - Matthew Heaney</td>
</tr>
<tr>
<td></td>
<td>T8</td>
<td>afternoon Architecture centered development and evolution of reliable real-time systems - Bruce Lewis, Ed Colbert</td>
</tr>
</tbody>
</table>

**Note:** No registration request will be confirmed until payment has been received. CANCELLATIONS must be in writing. A cancellation fee of 100 EUR will be applied to all cancellations. No refunds will be given for cancellations postmarked after May 2nd 2001. Substitutions will be accepted. The hotel information can be found through the web page of the conference; a hotel booking form is available on line also. Additional lunch tickets will be on sale throughout the conference.

For latest information see the web page at http://www.ada-europe.org/conference2001.html, or send email to ae2001-info@cs.kuleuven.ac.be.

**For any information, please contact:**

Esther Renson (Conference Secretariat)  
Department of Computer Science  
K.U.Leuven  
Celestijnenlaan 200A  
B-3001 Leuven  
Belgium  
Tel: ++32(0)16327640  
Fax ++32(0)16327996  
email: esther@cs.kuleuven.ac.be
Exhibiting

• Present your products to a world-wide audience at one of the major European Conferences in the field.
• Engage in discussions with leading researchers.
• Get access to state-of-the art research results.
• Build contacts for future interactions/consulting.
• Make a presentation at a full Conference session.
• Gain visibility, as the Ada-Europe’2001 web page will link to your website.

Exhibition space will be provided at the Maria Theresia College in the room where the coffee breaks will be held. The exhibition and a summary of the exhibits will be publicised in handouts, conference schedule, and conference program. Announcements will be made in the course of technical presentations.

Exhibitors will get:
• A stand in the coffee break area of the conference.
• A free conference registration for one person.
• A presentation of 30 minutes at a regular conference track.
• A link to your company homepage or to a conference specific web page from the Ada-Europe 2001 web page.

Sponsoring

A sliding scale of sponsorship provides a range of benefits. All levels include display of your logo on the conference website and in the program. The lowest level of support is very affordable!

Local sponsors enjoy
• exposure to an international audience;
• visibility of their support for local initiatives;
• visibility of their support for the local software industry and university.

Industry sponsors
• show visible commitment; saying that they are active and will remain active in this field.
• show commitment to Software Quality by publicly supporting a conference focusing on techniques for making good software.
• are seen helping with a major event in their community.

For more details about exhibiting or sponsoring, see: http://www.ada-europe.org/conference2001.html

Conference Chair
Karel De Vlaminck
K.U.Leuven
Department of Computer Science
Celestijnenlaan 200 A
B-3001 Leuven (Heverlee), Belgium
Karel.DeVlaminck@cs.kuleuven.ac.be

Program Co-Chairs
Dirk Craeynest
OFFIS nv/sa & K.U.Leuven
Weiveldaan 41/32
B-1930 Zaventem, Belgium
Dirk.Craeynest@cs.kuleuven.ac.be

Alfred Strohmeier
Swiss Fed. Inst. of Technology Lausanne
Software Engineering Lab
CH-1015 Lausanne EPFL,
Switzerland
Alfred.Strohmeier@epfl.ch

Tutorial Chair
Luc Bernard
OFFIS nv/sa, Belgium
lbn@offis.be

Exhibition Chair
Yvan Barbaix, K.U.Leuven
Phone +32-16-32.75.75
Fax +32-16-32.79.96
Yvan.Barbaix@cs.kuleuven.ac.be

Publicity Chair
Andrew Hately
Eurocontrol - CFMU, Belgium
Andrew.Hately@eurocontrol.be

Finance Co-Chairs
Karel De Vlaminck
K.U.Leuven, Belgium
Karel.DeVlaminck@cs.kuleuven.ac.be

Marc Gobin
Royal Military Academy, Belgium
Marc.Gobin@info.rma.ac.be

Local Organization Chair
Yolande Berbers
K.U.Leuven, Belgium
Yolande.Berbers@cs.kuleuven.ac.be

The organizers thank the exhibitors (preliminary list)

and the supporters of the conference (preliminary list)